An exact algorithm for the fixed charge transportation problem based on matching source and sink patterns

Mingozzi, Aristide; Roberti, Roberto

published in
Transportation Science
2018

DOI (link to publisher)
10.1287/trsc.2017.0742

document version
Publisher's PDF, also known as Version of record

document license
Article 25fa Dutch Copyright Act

Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

An Exact Algorithm for the Fixed Charge Transportation Problem Based on Matching Source and Sink Patterns

Aristide Mingozzi, Roberto Roberti

informs
http://pubsonline.informs.org/journal/trsc/

TRANSPORTATION SCIENCE
Vol. 52, No. 2, March–April 2018, pp. 229–238
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

An Exact Algorithm for the Fixed Charge Transportation Problem Based on Matching Source and Sink Patterns

Aristide Mingozzi,[a] Roberto Roberti[b]

[a] Department of Mathematics, University of Bologna, 40126 Bologna, Italy; [b] VU University Amsterdam, 1081 HV Amsterdam, Netherlands
Contact: aristide.mingozzi@unibo.it (AM); r.roberti@vu.nl, http://orcid.org/0000-0002-2987-1593 (RR)

Received: January 16, 2015
Revised: August 11, 2015; May 18, 2016
Accepted: September 28, 2016
Published Online in Articles in Advance: April 27, 2017

https://doi.org/10.1287/trsc.2017.0742

Abstract. This paper describes an exact algorithm for the fixed charge transportation problem based on a new integer programming formulation that involves two sets of variables representing flow patterns from sources to sinks and from sinks to sources. The formulation states to select a pattern for each source and each sink and to match the corresponding flows. The linear relaxation of this new formulation is enforced by adding a pseudo-polynomial number of equations that are shown to contain, as special cases, different valid inequalities recently proposed for the problem. The resulting lower bound dominates the lower bounds proposed in the literature. Such a lower bound is embedded into an exact branch-and-cut-and-price algorithm. Computational results on benchmark instances show that the proposed algorithm is several times faster than the state-of-the-art exact methods and could solve all open instances. New harder instances with up to 120 sources and 120 sinks were solved to optimality.

Supplemental Material: The online appendix is available at https://doi.org/10.1287/trsc.2017.0742.

Keywords: decomposition method • integer programming • column generation • branch and cut and price • fixed charge • transportation problem

1. Introduction

The *fixed charge transportation problem* (FCTP) is a generalization of the classical *transportation problem* and is defined on two sets of sources and sinks with integral positive supplies and demands, respectively. It is assumed that the cost for sending a nonzero quantity of commodity from each source to each sink equals a variable nonnegative cost proportional to the amount of commodity sent plus a fixed nonnegative cost. The FCTP asks to transport all quantities available at the sources to the sinks while minimizing the overall fixed and variable costs.

The FCTP is a special case of the *single commodity uncapacitated fixed charge network flow problem* (Rardin and Wolsey 1993, Ortega and Wolsey 2003), which itself is a special case of the more general *fixed charge problem* formulated by Hirsch and Dantzig (1968). Agarwal (2006) has shown that the FCTP is a special case of a large class of network design problems.

The FCTP becomes the *pure fixed charge transportation problem* (PFCTP) whenever only fixed costs are considered and all variable costs are zero (see Fisk and McKeown 1979 and Göthe-Lundgren and Larsson 1994). It is well known that both the FCTP and the PFCTP are $\mathcal{NP}$-hard.

The FCTP is a general model for several practical problems in distribution, transportation, scheduling, and location systems, where fixed costs represent toll charges on highways, landing fees at airports, setup costs in production systems, or costs for building roads. Real-life applications of the FCTP are described by Stroup (1967), Hirsch and Dantzig (1968), Walker (1976), Hultberg and Cardoso (1997), and Adlakha and Kowalski (2003).

Most of the exact methods proposed in the literature are based on a descriptive mixed-integer programming formulation with continuous variables to represent the flows from sources to sinks and binary variables to model the usage of the links between sources and sinks. Agarwal and Aneja (2012) studied the structure of the projection polyhedron of such a formulation in the space of its binary variables; they developed several classes of valid inequalities, which generalize the set covering inequalities, and derived conditions under which such inequalities are facet defining. Agarwal and Aneja (2012) also proposed an exact method able to solve randomly generated instances with up to 15 sources and 15 sinks.

Roberti, Bartolini, and Mingozzi (2015) introduced a new integer programming formulation with exponentially many variables corresponding to flow patterns from sources to sinks. They showed that the lower bound provided by the linear relaxation of this formulation is tighter than that given by the linear relaxation of the descriptive mixed-integer programming formulation. They also described different classes of valid inequalities to enforce such a lower bound and an exact branch-and-price algorithm. Computational results indicated that the proposed algorithm could solve instances with up to 70 sources and 70 sinks outperforming the previous exact algorithms from the literature.

Contribution of This Paper. This paper describes a new integer programming formulation for the FCTP with exponentially many variables and a polynomial number of constraints. Variables represent flow patterns from sources to sinks and flow patterns from sinks to sources. These two types of variables are matched together through a polynomial number of constraints to provide a valid FCTP solution. A family of equations are introduced to enforce the linear relaxation of the new formulation to achieve a lower bound that implies all valid inequalities introduced by Roberti, Bartolini, and Mingozzi (2015). It is shown that these latter inequalities correspond to different types of surrogate constraints of the new family of equations introduced in this paper. The achieved lower bound allows to roughly halve the gap left by the lower bound of Roberti, Bartolini, and Mingozzi (2015). This lower bound is then embedded into an exact branch-and-cut-and-price algorithm to obtain an optimal FCTP integer solution. Computational results on benchmark instances from the literature show that the proposed exact algorithm outperforms the previous exact algorithms from the literature as well as the recent method of Roberti, Bartolini, and Mingozzi (2015). The new algorithm is several times faster, can solve all instances previously unsolved, and can solve much harder FCTP instances with up to 120 sources and 120 sinks in reasonable computing times.

2. FCTP Formulations from the Literature

In this section, we survey the formulations used by the recent exact methods for the FCTP proposed by Agarwal and Aneja (2012) and by Roberti, Bartolini, and Mingozzi (2015).

The FCTP is defined on a complete bipartite graph $G = (S, T, A)$, where $S = \{1, 2, \ldots, m\}$ is a set of m sources and $T = \{1, 2, \ldots, n\}$ is a set of n sinks. At each source $i \in S$, an integer quantity $a_i > 0$, $a_i \in \mathbb{Z}_+$, of a commodity is available, and each sink $j \in T$ requires an integer quantity $b_j > 0$, $b_j \in \mathbb{Z}_+$, of the commodity from the sources. The arc set A is defined as $A = \{(i, j): i \in S, j \in T\}$. To send the commodity through arc $(i, j) \in A$, a unit cost c_{ij} plus a fixed cost f_{ij} is incurred. Without loss of generality, it is assumed that $\sum_{i \in S} a_i = \sum_{j \in T} b_j$.

2.1. The Descriptive Formulation Used by Agarwal and Aneja (2012)

The recent method of Agarwal and Aneja (2012) as well as most of the exact algorithms in the literature are based on the following well-known descriptive mixed-integer programming formulation, hereafter called F0.

Let x_{ij} be a continuous variable that represents the quantity of commodity sent from source i to sink j, and let y_{ij} be a (0-1) variable that equals 1 if and only if $x_{ij} > 0$ (0 otherwise). Let $m_{ij} = \min\{a_i, b_j\}$, for each arc $(i, j) \in A$. Formulation F0 is

[F0]
$$z^* = \min \sum_{(i,j) \in A} (c_{ij} x_{ij} + f_{ij} y_{ij}) \tag{1}$$
$$\text{s.t.} \quad \sum_{j \in T} x_{ij} = a_i, \quad i \in S, \tag{2}$$
$$\sum_{i \in S} x_{ij} = b_j, \quad j \in T, \tag{3}$$
$$0 \le x_{ij} \le m_{ij} y_{ij}, \quad (i, j) \in A, \tag{4}$$
$$y_{ij} \in \{0, 1\}, \quad (i, j) \in A. \tag{5}$$

We denote by LF0 the linear relaxation of problem F0 and by LB0 its optimal solution cost.

2.2. The Integer Programming Formulation of Roberti, Bartolini, and Mingozzi (2015)

The exact method presented by Roberti, Bartolini, and Mingozzi (2015) for the FCTP is based on the following integer programming formulation, hereafter called F1.

For each source $i \in S$, let $\mathscr{W}_i = \{\mathbf{w} \in \mathbb{Z}_+^n: \sum_{j \in T} w_j = a_i\}$. Let the vectors of the sets $\mathscr{W}_i$, $i \in S$, be indexed in such a way that $\mathscr{W}_i = \{l \in \mathbb{Z}: \sum_{s=1}^{i-1} |\mathscr{W}_s| < l \le \sum_{s=1}^{i} |\mathscr{W}_s|\}$ represents the index set of all vectors of the set $\mathscr{W}_i$. Any vector $\mathbf{w}^l$, $l \in \mathscr{W}_i$, is called *flow pattern* or, simply, *pattern* of source $i \in S$ and has an associated cost $d_l = \sum_{j \in T: w_j^l > 0} (c_{ij} w_j^l + f_{ij})$. Moreover, let $\mathscr{W}$ be the index set of all source patterns defined as $\mathscr{W} = \bigcup_{i \in S} \mathscr{W}_i$.

Let ξ_l be a (0-1) variable equal to 1 if and only if pattern $l \in \mathscr{W}$ is in solution (0 otherwise). Formulation F1 is

[F1]
$$z^* = \min \sum_{l \in \mathscr{W}} d_l \xi_l \tag{6}$$
$$\text{s.t.} \quad \sum_{l \in \mathscr{W}} w_j^l \xi_l = b_j, \quad j \in T, \tag{7}$$
$$\sum_{l \in \mathscr{W}_i} \xi_l = 1, \quad i \in S, \tag{8}$$
$$\xi_l \in \{0, 1\}, \quad l \in \mathscr{W}. \tag{9}$$

The objective function (6) aims to minimize the costs of the selected patterns. Constraints (7) state that all sink requests must be satisfied. Constraints (8) require that exactly one pattern for each source be chosen. Constraints (9) impose integrality on the variables.

Let LF1 be the linear relaxation of formulation F1 and $z(\text{LF1})$ its optimal solution cost.

Note that (see Roberti, Bartolini, and Mingozzi 2015) any LF1 solution $\boldsymbol{\xi}$ can be transformed into an LF0 solution $(\mathbf{x}, \mathbf{y})$ by setting

$$x_{ij} = \sum_{l \in \mathscr{W}_i} w_j^l \xi_l \quad \text{and} \quad y_{ij} = \sum_{l \in \mathscr{W}_i: w_j^l > 0} \xi_l, \quad \text{for each } (i, j) \in A. \tag{10}$$

Roberti, Bartolini, and Mingozzi (2015) showed that $z(\text{LF1}) \ge \text{LB0}$, and such an inequality can be strict.

3. The New Integer Programming Formulation

The new integer programming formulation proposed in this paper is based on the pattern sets $\mathcal{W}_i$, $i \in S$, associated with the sources, defined for formulation F1, and on the pattern sets $\bar{\mathcal{W}}_j$, $j \in T$, associated with the sinks and defined, similarly to $\mathcal{W}_i$, $i \in S$, as follows. Let $\bar{\mathcal{W}}_j$ be the set of all integer solutions of constraints (3) (i.e., $\bar{\mathcal{W}}_j = \{\bar{\mathbf{w}} \in \mathbb{Z}_+^m: \sum_{i\in S} \bar{w}_i = b_j\}$) for sink $j \in T$, and let $\bar{\mathcal{W}}_j = \{l \in \mathbb{Z}: \sum_{s=1}^{j-1} |\bar{\mathcal{W}}_s| < l \le \sum_{s=1}^{j} |\bar{\mathcal{W}}_s|\}$ be the index set of all vectors of the set $\bar{\mathcal{W}}_j$. Any vector $\bar{\mathbf{w}}^l$, $l \in \bar{\mathcal{W}}_j$, is called *flow pattern* or, simply, *pattern* of sink $j \in T$ and has an associated cost $\bar{d}_l = \sum_{i\in S:\bar{w}_i^l>0}(c_{ij}\bar{w}_i^l + f_{ij})$. Moreover, let $\bar{\mathcal{W}}$ be the index set of all sink patterns defined as $\bar{\mathcal{W}} = \bigcup_{j\in T} \bar{\mathcal{W}}_j$.

In addition to the binary variables ξ_l, $l \in \mathcal{W}$, defined for formulation F1 in Section 2.2, let $\bar{\xi}_l$ be a (0-1) variable associated to each pattern $l \in \bar{\mathcal{W}}$ that equals 1 if and only if pattern l is in solution (0 otherwise). The new formulation F2 is

$$\text{[F2]} \quad z^* = \frac{1}{2}\min\left\{\sum_{l\in\mathcal{W}} d_l\xi_l + \sum_{l\in\bar{\mathcal{W}}} \bar{d}_l\bar{\xi}_l\right\} \tag{11}$$

$$\text{s.t.} \quad \sum_{l\in\mathcal{W}} d_l\xi_l - \sum_{l\in\bar{\mathcal{W}}} \bar{d}_l\bar{\xi}_l = 0, \tag{12}$$

$$\sum_{l\in\mathcal{W}_i} \xi_l = 1, \quad i \in S, \tag{13}$$

$$\sum_{l\in\bar{\mathcal{W}}_j} \bar{\xi}_l = 1, \quad j \in T, \tag{14}$$

$$\sum_{l\in\mathcal{W}_i} w_j^l\xi_l - \sum_{l\in\bar{\mathcal{W}}_j} \bar{w}_i^l\bar{\xi}_l = 0, \quad (i,j) \in A, \tag{15}$$

$$\xi_l \in \{0,1\}, \quad l \in \mathcal{W}, \tag{16}$$

$$\bar{\xi}_l \in \{0,1\}, \quad l \in \bar{\mathcal{W}}. \tag{17}$$

The objective (11) aims to minimize the total cost of the selected patterns. Constraint (12) imposes that the total cost of the selected source patterns is equal to the cost of the selected sink patterns. Constraints (13) and (14) force selection of one pattern $l \in \mathcal{W}_i$ for each source $i \in S$ and one pattern $l \in \bar{\mathcal{W}}_j$ for each sink $j \in T$. Constraints (15) state that if a pattern $l \in \mathcal{W}_i$ sending a flow w_j^l to sink $j \in T$ is selected, then a pattern $l \in \bar{\mathcal{W}}_j$ sending a flow $\bar{w}_i^l$ from source i must be selected.

It is worth noting that constraint (12) is redundant but not when integrality constraints (16)–(17) are relaxed.

We denote by LF2 the linear relaxation of formulation F2 and by LB2 its optimal solution cost.

Any LF2 solution $(\xi, \bar{\xi})$ can be transformed into an LF0 solution $(\mathbf{x}, \mathbf{y})$ having the same cost either by using expressions (10) or, equivalently because of Equation (15), by setting

$$x_{ij} = \sum_{l\in\bar{\mathcal{W}}_j} \bar{w}_i^l\bar{\xi}_l \quad \text{and} \quad y_{ij} = \sum_{l\in\bar{\mathcal{W}}_j:\bar{w}_i^l>0} \bar{\xi}_l, \quad \text{for each } (i,j) \in A. \tag{18}$$

Lemma 1. *Any* LF2 *solution* $(\xi, \bar{\xi})$ *corresponds to a feasible* LF1 *solution* ξ.

Proof. Solution ξ obviously satisfies constraints (8). To show that ξ satisfies constraints (7), too, it is sufficient to observe that these latter equations are surrogate constraints of Equation (15). By adding Equation (15) over all arcs (i,j), $i \in S$, for a given sink $j \in T$, we obtain $\sum_{i\in S}\sum_{l\in\mathcal{W}_i} w_j^l\xi_l - \sum_{l\in\bar{\mathcal{W}}_j}\sum_{i\in S}\bar{w}_i^l\bar{\xi}_l = 0$. Because $\sum_{i\in S}\sum_{l\in\mathcal{W}_i} w_j^l\xi_l = \sum_{l\in\mathcal{W}} w_j^l\xi_l$ and because, by definition, $\sum_{i\in S}\bar{w}_i^l = b_j$, the previous equation becomes $\sum_{l\in\mathcal{W}} w_j^l\xi_l - b_j\sum_{l\in\bar{\mathcal{W}}_j}\bar{\xi}_l = 0$ that, because of Equation (14), corresponds to constraints (7) for sink $j \in T$. □

4. The Exact Methods of Agarwal and Aneja (2012) and Roberti, Bartolini, and Mingozzi (2015)

In this section, we briefly survey the exact methods of Agarwal and Aneja (2012) and Roberti, Bartolini, and Mingozzi (2015) based on formulations F0 and F1, respectively.

4.1. The Exact Method of Agarwal and Aneja (2012)

Agarwal and Aneja (2012) studied the structure of the projection polyhedron of formulation F0 in the space of variables y_{ij}. They developed several classes of valid inequalities, which generalize the set covering inequalities, and derived conditions under which such inequalities are facet defining. Their exact method could solve randomly generated instances of both FCTP and PFCTP with up to 15 sources and 15 sinks.

Agarwal and Aneja (2012) improved the lower bound LB0 by adding, to LF0, valid inequalities based on the generalization of the following *set covering* (SC) inequalities originally proposed by Aneja (1974) and used by Göthe-Lundgren and Larsson (1994)

$$\text{(SC)} \qquad \sum_{i\in S\setminus K}\sum_{j\in L} y_{ij} \ge 1, \quad K \subset S,\ L \subset T: \quad \sum_{i\in K} a_i < \sum_{j\in L} b_j. \tag{19}$$

Agarwal and Aneja (2012) have shown that the SC inequalities are either facet defining, or that, under two mild conditions, they can be lifted to make them facet defining. These latter inequalities are called second order facets. They also showed that inequalities (19) can be separated exactly by solving a binary problem that looks for a subset of sources K and a subset of sinks L, such that $\sum_{i\in K} a_i < \sum_{j\in L} b_j$, having a minimum flow along the arcs of the cut set $\{(i,j) \in A: i \in K, j \in L\}$; if such a flow is less than 1, then a violated SC inequality (19) is found. The FCTP is then solved using the branch-and-cut feature of Cplex, where at each node of the branch-and-bound tree, separation heuristics are used to generate and add more cuts to the subproblem at that node.

4.2. The Exact Method of Roberti, Bartolini, and Mingozzi (2015)

Roberti, Bartolini, and Mingozzi (2015) described a branch-and-price algorithm where the lower bound at each node is computed by solving, through column generation, a relaxation, hereafter called $\overline{\text{LF1}}$, obtained by adding, to LF1, different types of valid inequalities.

The pricing problem for generating patterns of negative reduced cost for each source $i \in S$ corresponds to a *multiple choice knapsack problem* (MCKP) that is solved by a dynamic programming recursion in time $O(na_i^2)$.

4.2.1. Lower Bound LB1. Because the FCTP is symmetric in the sources and sinks, variables $\boldsymbol{\xi}$ could have been defined for sink patterns instead of source patterns leading to another formulation of the problem. This latter formulation, hereafter called $\overleftarrow{\text{F1}}$, corresponds to formulation F1 where sources and sinks are interchanged. Let $\overleftarrow{\text{LF1}}$ be the linear relaxation of $\overleftarrow{\text{F1}}$ and $z(\overleftarrow{\text{LF1}})$ its optimal solution cost. A valid lower bound, hereafter called LB1, is obtained by solving both $\overleftarrow{\text{LF1}}$ and LF1 and by setting $\text{LB1} = \max\{z(\overleftarrow{\text{LF1}}), z(\text{LF1})\}$.

4.2.2. Improved Lower Bound $\overline{\text{LB1}}$ Based on Valid Inequalities. To improve the lower bound LB1, Roberti, Bartolini, and Mingozzi (2015) added, to LF1 and $\overleftarrow{\text{LF1}}$, an adaptation of SC inequalities (19) and several classes of new valid inequalities having the property that their dual variables do not modify the constraints of the pricing problem.

We denote by $\overline{\text{LB1}}$ the value of the lower bound obtained by strengthening the lower bound LB1 with the following two classes of valid inequalities.

- *Inequalities of Class* 1. The first class of inequalities corresponds to a family H_j of different types of inequalities for constraint (7) of a given sink $j \in T$ that are satisfied by any integer F1 solution but that can be violated by a fractional LF1 solution. These inequalities, called "extended generalized upper bound cover," "couple," "feasibility," "Chvátal–Gomory," and "lifted Chvátal–Gomory," are briefly described in the appendix.
- *Inequalities of Class* 2. These inequalities correspond to the SC inequalities (19) where variables y_{ij} are substituted with variables ξ_l using expressions (10), namely

$$\text{(SC)} \qquad \sum_{i \in S \setminus K} \sum_{l \in \mathcal{W}_i} \tau_l(L)\xi_l \geq 1, \quad K \subset S,\ L \subset T: \sum_{i \in K} a_i < \sum_{j \in L} b_j, \tag{20}$$

where $\tau_l(L) = |\{j \in L: w_j^l > 0\}|$. Using relationship (18), any LF2 solution can be converted into a feasible LF0 solution, so the separation of violated inequalities (20) can be done in the space of the $\mathbf{y}$ variables as proposed by Agarwal and Aneja (2012), and any pair of subsets K and L violating inequalities (19) provides a violated inequality (20).

5. New Lower Bounds from Relaxation LF2

In this section, we first show that the lower bound LB2 dominates the lower bound LB1 derived from the formulation F1. Then, we describe new valid inequalities to strengthen the relaxation $\overline{\text{LF2}}$ and show that the resulting lower bound, called $\overline{\text{LB2}}$, dominates the lower bound obtained by adding, to LF1 and $\overleftarrow{\text{LF1}}$, only inequalities of Class 1. Finally, we described a bounding procedure to compute both the lower bound LB2 and the lower bound $\overline{\text{LB2}}$.

5.1. Comparing Lower Bounds LB2 and LB1

The following proposition states the relationship between lower bounds LB2 and LB1.

Proposition 1. *The inequality* $\text{LB2} \geq \text{LB1}$ *holds.*

Proof. Let $(\boldsymbol{\xi}, \bar{\boldsymbol{\xi}})$ be an optimal LF2 solution of cost LB2. According to Lemma 1, $\boldsymbol{\xi}$ corresponds to a feasible (not necessarily optimal) LF1 solution of cost $\bar{z}(\text{LF1}) = \sum_{l \in \mathcal{W}} d_l \xi_l \geq z(\text{LF1})$. Similarly, $\bar{\boldsymbol{\xi}}$ corresponds to an $\overleftarrow{\text{LF1}}$ solution of cost $\bar{z}(\overleftarrow{\text{LF1}}) = \sum_{l \in \bar{\mathcal{W}}} \bar{d}_l \bar{\xi}_l \geq z(\overleftarrow{\text{LF1}})$. Because of expression (11) and Equation (12), we have $\text{LB2} = \bar{z}(\text{LF1}) = \bar{z}(\overleftarrow{\text{LF1}})$, so $\text{LB2} \geq \text{LB1} = \max\{z(\text{LF1}), z(\overleftarrow{\text{LF1}})\}$. □

5.2. Lower Bound $\overline{\text{LB2}}$

Lower bound LB2 can be improved by adding, to LF2, the following pseudopolynomial number of equations.

For each arc $(i,j) \in A$ and each $q = 1, \ldots, m_{ij}$, let $\mathcal{W}_i(j,q) \subseteq \mathcal{W}_i$ be the set of patterns of source i delivering q units of commodity to destination j (i.e., $\mathcal{W}_i(j,q) = \{l \in \mathcal{W}_i: w_j^l = q\}$), and let $\bar{\mathcal{W}}_j(i,q) \subseteq \bar{\mathcal{W}}_j$ be the set of patterns of sink j sending a flow q from source i to destination j (i.e., $\bar{\mathcal{W}}_j(i,q) = \{l \in \bar{\mathcal{W}}_j: \bar{w}_i^l = q\}$). The following $\sum_{(i,j) \in A} m_{ij}$ equations, hereafter called *arc-quantity* (AQ) equations, are valid for relaxation LF2

$$\text{(AQ)} \qquad \sum_{l \in \mathcal{W}_i(j,q)} \xi_l - \sum_{l \in \bar{\mathcal{W}}_j(i,q)} \bar{\xi}_l = 0, \quad (i,j) \in A\ q = 1, \ldots, m_{ij}. \tag{21}$$

Equation (21) states that, for each arc $(i,j) \in A$ and quantity $q = 1, \ldots, m_{ij}$, any feasible FCTP solution containing a variable $\xi_l = 1$, $l \in \mathcal{W}_i$, such that $w_j^l = q$, must also contain a variable $\bar{\xi}_l = 1$, $l \in \bar{\mathcal{W}}_j$, such that $\bar{w}_i^l = q$. The (AQ) Equation (21) can be exactly separated by inspection in pseudopolynomial time.

The effectiveness of (AQ) equations is shown by the following proposition.

Proposition 2. *Each inequality of Class* 1 *introduced by Roberti, Bartolini, and Mingozzi* (2015) *for relaxation* LF1 *corresponds to a different surrogate constraint of the* (AQ) *Equation* (21).

Proof. See the online appendix. □

We denote by $\overline{\mathrm{LF2}}$ the relaxation obtained by adding the (AQ) Equation (21) to LF2 and by $\overline{\mathrm{LB2}}$ the optimal $\overline{\mathrm{LF2}}$ solution cost.

Corollary 1. *Lower bound* $\overline{\mathrm{LB2}}$ *is greater than or equal to the lower bound obtained by adding, to relaxations* LF1 *and* $\overleftarrow{\mathrm{LF1}}$, *the inequalities of Class* 1 *but not the inequalities of Class* 2.

Proof. It follows from Proposition 2. □

We should mention that an optimal $\overline{\mathrm{LF2}}$ solution can violate inequalities (20). Nonetheless, from our computational experience that will be summarized in Section 7, it is not worth separating and adding such inequalities to $\overline{\mathrm{LF2}}$ because they were violated in only a few instances and the overall performance of the exact algorithm that will be described in Section 6 does not benefit from their addition. This is because the separation time is relevant and the lower bound is usually tightened by less than 0.01%.

5.3. A Bounding Procedure to Compute Lower Bounds LB2 and $\overline{\mathrm{LB2}}$

The bounding procedure we propose first solves relaxation LF2 by column generation to compute the lower bound LB2. Second, it solves $\overline{\mathrm{LF2}}$ by iteratively adding, in a cut-and-column generation fashion, the (AQ) Equation (21) and patterns of negative reduced cost. The bounding procedure terminates as soon as no negative reduced cost patterns exist and no (AQ) equations are violated and provides the lower bound $\overline{\mathrm{LB2}}$.

At a given iteration, let $\alpha \in \mathbb{R}$, $u_i \in \mathbb{R}$ $(i \in S)$, $v_j \in \mathbb{R}$ $(j \in T)$, $g_{ij} \in \mathbb{R}$ $((i,j) \in A)$, and $h_{ijq} \in \mathbb{R}$ $((i,j) \in A$, $q = 1, \ldots, m_{ij})$ be the dual variables of the master problem (MP) associated with constraints (12)–(15), and (21), respectively. The pricing problem consists of finding the pattern of each set $\mathcal{W}_i$, $i \in S$, and each set $\overleftarrow{\mathcal{W}}_j$, $j \in T$, having the most negative reduced costs with respect to the current dual vectors $(\alpha, \mathbf{u}, \mathbf{v}, \mathbf{g}, \mathbf{h})$. In the following, we describe the pricing problem for each source $i \in S$. It is obvious that a similar method can be used to solve the pricing for each sink $j \in T$.

The pricing problem is similar to that described by Roberti, Bartolini, and Mingozzi (2015), but it differs in the way the pattern reduced cost is computed. In particular, the reduced cost (say, d'_l) of pattern $l \in \mathcal{W}_i$, $i \in S$, is

$$d'_l = \sum_{j \in T: w^l_j > 0} ((1-\alpha)(f_{ij} + c_{ij} w^l_j) - w^l_j g_{ij} - h_{ijw^l_j}) - u_i.$$

The reduced cost d'_l can be spread throughout the arcs $(i,j) \in A$, $j \in T$, such that $w^l_j > 0$ as follows. Define the modified cost δ_{ijq}, with respect to the dual solution $(\alpha, \mathbf{u}, \mathbf{v}, \mathbf{g}, \mathbf{h})$, for sending a quantity q $(1 \le q \le m_{ij})$ along arc $(i,j) \in A$ as $\delta_{ijq} = (1-\alpha) f_{ij} + ((1-\alpha)c_{ij} - g_{ij})q - h_{ijq}$. Then, d'_l can equivalently be written as $d'_l = \sum_{j \in T: w^l_j > 0} \delta_{ijw^l_j} - u_i$. Therefore, the pricing problem for source $i \in S$ consists of solving the following *multiple choice knapsack problem* (MCKP(i)) where the binary variable φ_{jq} $(j \in T$, $q = 1, \ldots, m_{ij})$ equals 1 if q units of commodity are sent from source i to sink j (and equals 0 otherwise):

(MCKP(i))

$$z_i = \min \sum_{j \in T} \sum_{q=1}^{m_{ij}} \delta_{ijq} \varphi_{jq} - u_i \tag{22}$$

$$\text{s.t.} \quad \sum_{j \in T} \sum_{q=1}^{m_{ij}} q \varphi_{jq} = a_i, \tag{23}$$

$$\sum_{q=1}^{m_{ij}} \varphi_{jq} \le 1, \quad j \in T, \tag{24}$$

$$\varphi_{jq} \in \{0,1\}, \quad j \in T,\ q = 1, \ldots, m_{ij}. \tag{25}$$

Let $\boldsymbol{\varphi}^*$ be the optimal (MCKP(i)) solution. If $z_i < 0$, then the pattern $l^* \in \mathcal{W}_i$ defined as $w^{l^*}_j = \sum_{q=1}^{m_{ij}} q \varphi^*_{jq}$, $j \in T$, has a negative reduced cost.

As described by Roberti, Bartolini, and Mingozzi (2015), the problem (MCKP(i)) can be solved by dynamic programming in time $O(na_i^2)$ as follows. Let $f(\sigma, j)$ be the optimal solution of the subproblem derived from problem (MCKP(i)) by replacing the right-hand side a_i of Equation (23) with σ $(1 \le \sigma \le a_i)$ and the set T in constraints (24) with $\{1, 2, \ldots, j\}$. The recursion for computing functions $f(\sigma, j)$ for each $j = 1, \ldots, n$ and $\sigma = 1, \ldots, \min\{a_i, \sum_{s=1}^{j} b_j\}$ is

$$f(\sigma, j) = \min\left\{ f(\sigma, j-1), \min_{1 \le \sigma' \le \min\{m_{ij}, \sigma\}} \{ f(\sigma - \sigma', j-1) + \delta_{ij\sigma'} \} \right\}.$$

The initialization $f(0, j) = -u_i$, $j \in T$, and $f(\sigma, 0) = \infty$, $\sigma = 1, \ldots, a_i$, is required. The cost z_i of the pattern of minimum reduced cost corresponds to the value of function $f(a_i, n)$.

6. Branch-and-Cut-and-Price Algorithm

The new exact branch-and-cut-and-price algorithm proposed in this paper executes the bounding procedure described in Section 5.3 to compute the lower bound at the root node. Whenever the optimal solution of the master problem is not a feasible F2 solution, branching is applied. Nodes are explored with the best-bound-first strategy. Branching is on y_{ij} variables computed according to expressions (10). Given a fractional LF2 solution, the disjunction $y_{ij} = 0 \vee y_{ij} = 1$ is imposed on the variable y_{ij} having the value closest to 0.6. Ties are broken by selecting the arc having the largest fixed cost f_{ij}.

The master problem solved at each node is problem LF2 plus a subset of Equation (21). The lower bound computed corresponds to the lower bound $\overline{\mathrm{LB2}}$, where the points made infeasible by the decisions made on the y_{ij} variables at the ancestors nodes are removed from the sets A. The lower bound is computed with a cut-and-column generation algorithm that generates negative-reduced-cost columns with the method described in Section 5.3 and adds violated (AQ) Equation (21). The initial master problem contains the columns (feasible for the current node) having a strictly positive value in the optimal primal solutions of the father node plus the columns corresponding to a greedy solution of the master problem of the current node to guarantee that the master problem contains a feasible primal solution. The initial set of (AQ) Equation (21) contained in the master problem are inherited from the father node.

7. Computational Results

This section reports the computational results achieved by the exact algorithm (hereafter called MR) described in Section 6 and compares its performance with the branch-and-cut algorithm of Agarwal and Aneja (2012) (hereafter AA), the branch-and-price algorithm of Roberti, Bartolini, and Mingozzi (2015) (hereafter RBM), and Cplex 12.5 solving formulation F0. Algorithm MR was coded in C and compiled with Visual Studio 2010 64-bit. The master problem (MP) of the bounding procedure described in Section 5.3 was solved by using the LP-solver of Cplex 12.5.

All computational experiments concerning Cplex 12.5, RBM, and MR were conducted on an Intel Xeon X7350 at 2.93 GHz server with 16 GB RAM running on a single core. Algorithm AA was tested on an Intel Pentium 1.8 GHz Dual Core CPU (and by using Cplex 11.2 as the LP-solver), which, according to SPEC[1], is about half the speed of the Intel Xeon X7350 at 2.93 GHz. All computing times reported later in this section are in seconds.

A total of 463 instances divided into four sets (hereafter called Data Set 1, Data Set 2, Data Set 3, and Data Set 4) were considered:

• Data Set 1 is available at http://plato.asu.edu/ftp/lptestset/fctp (accessed December 15, 2016), is maintained by Arizona State University, and is made up of 13 instances with up to 17 sources and 64 sinks. These instances were used to test both AA and RBM.

• Data Set 2, Data Set 3, and Data Set 4 are divided into 3, 18, and 18 classes, respectively, each one containing 10 instances. All 390 instances feature $m = n$ (i.e., same number of sources and sinks), fixed costs f_{ij} randomly generated in the interval $[200, 800]$, and values a_i and b_j randomly generated in the interval $[1, B]$. Each class is characterized by a different combination of values n and B and by the way unit costs c_{ij} are generated.

—Data Set 2 was introduced by Agarwal and Aneja (2012) and consists of three classes featuring $n = 15$, $B = 20$, and random unit costs properly scaled so as to maintain a predefined ratio θ (with $\theta = 0.0, 0.2, 0.5$) between the total variable and fixed costs in the optimal solution.

—Data Set 3 was introduced by Roberti, Bartolini, and Mingozzi (2015) and consists of 18 classes. Unit costs c_{ij} are computed as $c_{ij} = \lfloor(\theta f_{ij}(2n-1))/\sum_{i\in O} a_i\rfloor$. To build 18 classes, six combinations of (n, B) were considered (i.e., (30, 20), (50, 20), (70, 20), (20, 50), (30, 50), (40, 50)) and three values of θ (i.e., $\theta = 0.0, 0.2, 0.5$).

—Data Set 4 is made up of 240 new instances generated as the instances of Data Set 3 but with different (n, B) combinations. The eight combinations (100, 20), (120, 20), (50, 50), (60, 50), (70, 50), (40, 100), (50, 100), and (60, 100) were considered; for each combination, three classes were generated by varying the setting of value $\theta = 0.0, 0.2, 0.5$.

All instances considered fulfill the initial assumption that the sum of the supplies available at the sources is equal to the sum of the requests of the sinks (i.e., $\sum_{i\in S} a_i = \sum_{j\in T} b_j$). When this condition does not hold, a dummy source or a dummy sink needs to be added. As for the exact method of Roberti, Bartolini, and Mingozzi (2015), we have not seen any significant change in the performance of the exact algorithm proposed in this paper when using unbalanced instead of balanced instances, so we limit our computational study to balanced instances only, as commonly done in the literature about the FCTP.

Tables 1–7 compare the computational performance of the four algorithms Cplex 12.5, AA, RBM, and MR. Columns reported in these tables have the following meaning:

• Inst: instance name;
• n (θ): setting of n (θ);
• z*: optimal solution cost;
• LB0 (LB1, $\overline{\text{LB1}}$, LB2, $\overline{\text{LB2}}$): percentage gap left by the lower bound LB0 (LB1, $\overline{\text{LB1}}$, LB2, $\overline{\text{LB2}}$, respectively);
• $\overline{\text{LB0}}$ under heading Cplex 12.5: percentage gap left by the lower bound achieved at the root node by Cplex 12.5 after adding all its cuts to problem LF0;
• $\overline{\text{LB0}}$ under heading AA: percentage gap left by the lower bound achieved at the root node by AA after adding, to problem LF0, all Cplex 11.2 cuts and different set covering inequalities of Agarwal and Aneja (2012);
• Cut (Col): number of (AQ) Equation (21) (columns) generated at the root node;
• Nd: number of nodes of the search tree;
• $T_{\overline{\text{LB1}}}$ ($T_{\overline{\text{LB2}}}$): time spent for computing the lower bound $\overline{\text{LB1}}$ ($\overline{\text{LB2}}$);
• T_{MP} (T_{Pr}): computing time spent for solving the master (pricing) problem;
• T_{Tot}: total computing time;
• Opt: number of instances solved to optimality.

Table 1. Data Set 1

		Cplex 12.5			AA	RBM					MR								
Inst.	z^*	LB0	$\overline{\text{LB0}}$	T_{Tot}	T_{Tot}	LB1	$\overline{\text{LB1}}$	$T_{\overline{LB1}}$	Nd	T_{Tot}	LB2	$\overline{\text{LB2}}$	$T_{\overline{LB2}}$	Cut	Col	Nd	T_{MP}	T_{Pr}	T_{Tot}
4×64	9,711	0.8	0.2	0.2	1.1	0.2	0.0	1.9	0	1.9	0.2	0.1	3.7	21	2,510	2	9.0	1.1	10.2
8×32	5,247	5.9	1.4	2.6	11.0	3.0	0.0	0.3	0	0.3	2.3	0.0	0.4	175	1,028	0	0.3	0.1	0.4
10×10a	1,499	16.4	0.5	0.3	0.4	10.0	0.0	0.1	0	0.1	6.9	0.0	0.1	79	339	0	0.0	0.0	0.1
10×10b	3,073	14.9	2.1	0.3	0.7	8.4	0.0	0.1	0	0.1	3.7	0.0	0.1	65	238	0	0.0	0.0	0.1
10×10c	13,007	13.9	4.0	0.8	1.9	8.9	0.8	0.4	17	0.5	4.4	1.6	0.1	79	304	10	0.2	0.0	0.2
10×12	2,714	10.6	1.3	0.2	0.6	5.6	0.0	0.1	0	0.1	2.1	0.0	0.0	41	268	0	0.0	0.0	0.0
10×26	4,270	9.6	4.0	12.3	22.4	4.7	0.6	1.0	7	2.2	4.7	0.7	0.6	291	1,307	10	1.1	0.2	1.3
12×12	2,291	20.3	6.8	5.4	5.6	12.5	1.4	0.9	23	1.0	10.0	0.7	0.2	183	581	4	0.2	0.1	0.3
12×21	3,664	13.8	5.7	36.2	19.3	8.5	1.1	0.5	35	1.0	7.2	0.4	0.3	273	1,080	8	0.8	0.1	0.8
13×13	3,252	17.2	6.4	9.0	10.4	12.3	1.4	0.8	55	1.2	10.1	0.5	0.2	191	714	2	0.2	0.0	0.3
14×18	3,712	18.7	9.0	818.8	1,307.5	11.9	1.4	0.5	81	2.0	10.3	0.8	0.5	444	1,171	16	1.5	0.3	1.8
16×16	3,823	18.5	6.9	91.9	83.7	11.8	1.8	1.9	145	4.0	9.0	0.8	0.3	271	772	14	0.8	0.1	0.9
17×17	1,373	11.4	1.7	2.6	5.5	6.9	0.0	1.1	0	1.1	5.8	0.0	0.2	198	834	0	0.2	0.0	0.2
		13.2	3.8	75.4	113.1	8.0	0.7	0.7	28	1.2	5.9	0.4	0.5	178	857	5	1.1	0.2	1.3

Table 2. Data Set 2 (Instances with $n = 15$, $B = 20$)

	Cplex 12.5				AA			RBM						MR									
θ	LB0	$\overline{\text{LB0}}$	T_{Tot}	Opt	$\overline{\text{LB0}}$	T_{Tot}	Opt	LB1	$\overline{\text{LB1}}$	$T_{\overline{LB1}}$	Nd	T_{Tot}	Opt	LB2	$\overline{\text{LB2}}$	$T_{\overline{LB2}}$	Cut	Col	Nd	T_{MP}	T_{Pr}	T_{Tot}	Opt
0.0	25.1	7.1	20.3	10	10.4	139.0	10	16.0	0.5	0.7	9	0.8	10	11.2	0.1	0.1	222	645	2	0.2	0.0	0.2	10
0.2	18.1	4.7	36.9	10	7.5	51.3	8	11.8	0.4	0.4	11	0.5	10	7.1	0.1	0.1	190	654	2	0.2	0.0	0.2	10
0.5	14.3	3.6	18.2	10	6.4	61.6	9	7.7	0.3	0.5	12	0.6	10	6.0	0.2	0.1	179	649	2	0.2	0.0	0.2	10
	19.2	5.1	25.1	30	8.1	83.9	27	11.8	0.4	0.5	11	0.6	30	8.1	0.1	0.1	197	650	2	0.2	0.0	0.2	30

Table 3. Data Set 3 (Instances with $B = 20$)

		Cplex 12.5				RBM						MR									
n	θ	LB0	$\overline{\text{LB0}}$	T_{Tot}	Opt	LB1	$\overline{\text{LB1}}$	$T_{\overline{LB1}}$	Nd	T_{Tot}	Opt	LB2	$\overline{\text{LB2}}$	$T_{\overline{LB2}}$	Cut	Col	Nd	T_{MP}	T_{Pr}	T_{Tot}	Opt
30	0.0	18.2	5.8	3,221	4	11.6	0.9	9	234	13	10	6.8	0.6	0	332	1,245	28	2	0	2	10
30	0.2	15.3	5.0	4,784	6	9.6	0.6	7	176	11	10	6.3	0.5	0	325	1,211	27	2	0	2	10
30	0.5	12.9	3.8	1,850	8	8.1	0.5	6	90	9	10	5.6	0.4	0	313	1,131	19	2	0	2	10
50	0.0	17.2	5.3	—	0	10.3	0.7	39	2,110	116	10	5.7	0.6	1	524	2,179	161	23	3	26	10
50	0.2	14.5	5.0	—	0	9.4	0.6	45	1,109	95	10	5.6	0.4	1	568	2,225	105	19	2	22	10
50	0.5	11.3	3.7	—	0	6.7	0.5	40	7,489	324	10	4.5	0.3	1	558	2,207	245	41	5	47	10
70	0.0	15.0	4.7	—	0	9.3	0.5	181	3,251	421	10	4.7	0.4	3	674	3,357	244	67	10	80	10
70	0.2	13.2	4.4	—	0	8.6	0.5	119	12,830	1,055	9	4.5	0.4	3	775	3,255	525	157	22	184	10
70	0.5	10.4	3.6	—	0	6.7	0.4	140	14,306	1,395	10	3.8	0.3	3	781	3,123	500	163	21	190	10
		14.2	4.6	3,285	18	8.9	0.6	65	4,622	382	89	5.3	0.4	2	539	2,215	206	53	7	62	90

Table 4. Data Set 3 (Instances with $B = 50$)

		Cplex 12.5				RBM						MR									
n	θ	LB0	$\overline{\text{LB0}}$	T_{Tot}	Opt	LB1	$\overline{\text{LB1}}$	$T_{\overline{LB1}}$	Nd	T_{Tot}	Opt	LB2	$\overline{\text{LB2}}$	$T_{\overline{LB2}}$	Cut	Col	Nd	T_{MP}	T_{Pr}	T_{Tot}	Opt
20	0.0	26.4	10.2	4,569	4	16.9	1.9	2	1,306	37	10	12.3	1.0	1	590	1,457	25	3	0	4	10
20	0.2	20.8	7.2	2,269	7	13.8	1.5	3	444	16	10	9.8	0.7	1	503	1,294	11	2	0	2	10
20	0.5	15.9	5.1	1,234	10	10.0	1.0	3	282	13	10	7.3	0.3	1	478	1,332	6	1	0	1	10
30	0.0	23.4	9.6	—	0	16.2	1.7	11	5,497	353	10	10.9	0.9	2	908	2,287	62	15	2	18	10
30	0.2	22.0	9.4	—	0	15.0	1.7	9	11,264	899	10	10.8	0.8	2	993	2,390	74	21	3	25	10
30	0.5	16.1	6.4	—	0	11.0	1.3	8	5,676	387	10	8.2	0.5	2	882	2,314	49	11	2	13	10
40	0.0	22.3	9.5	—	0	15.7	1.9	27	51,214	5,060	5	10.2	1.0	5	1,305	3,413	377	137	23	163	10
40	0.2	19.6	7.9	—	0	12.8	1.5	24	17,418	1,753	6	9.0	0.9	4	1,222	3,134	587	273	36	315	10
40	0.5	15.2	6.4	—	0	10.2	1.3	25	13,870	1,741	3	7.5	0.7	5	1,248	3,299	278	121	19	143	10
		20.2	8.0	2,690	21	13.5	1.5	13	11,886	1,140	74	9.5	0.8	2	903	2,324	163	65	10	76	90

Table 5. Data Set 4 (Instances with $B = 20$)

		Cplex 12.5			MR									
n	θ	LB0	$\overline{\text{LB0}}$	Opt	LB2	$\overline{\text{LB2}}$	$T_{\overline{LB2}}$	Cut	Col	Nd	T_{MP}	T_{Pr}	T_{Tot}	Opt
100	0.0	13.6	3.8	0	3.5	0.4	7	867	4,856	1,706	870	146	1,048	10
100	0.2	12.2	3.8	0	3.9	0.3	9	1,033	5,297	2,703	1,626	235	1,907	10
100	0.5	10.5	3.4	0	3.8	0.3	9	1,035	5,295	2,306	1,808	237	2,089	10
120	0.0	12.8	3.6	0	3.1	0.3	10	1,061	6,008	5,309	3,706	655	4,484	10
120	0.2	11.9	4.0	0	3.8	0.5	15	1,266	6,715	5,030	3,160	490	3,743	9
120	0.5	10.3	3.5	0	3.7	0.3	16	1,347	6,865	7,639	7,927	1,097	9,239	10
		11.9	3.7	0	3.7	0.3	11	1,102	5,839	4,116	3,183	477	3,752	59

Table 6. Data Set 4 (Instances with $B = 50$)

		Cplex 12.5			MR									
n	θ	LB0	$\overline{\text{LB0}}$	Opt	LB2	$\overline{\text{LB2}}$	$T_{\overline{LB2}}$	Cut	Col	Nd	T_{MP}	T_{Pr}	T_{Tot}	Opt
50	0.0	22.0	9.7	0	10.1	0.9	10	1,734	4,590	2,154	1,100	176	1,303	10
50	0.2	17.7	7.4	0	7.9	0.7	8	1,582	4,287	776	478	83	572	10
50	0.5	14.5	5.9	0	6.7	0.6	9	1,434	4,623	527	312	55	374	10
60	0.0	20.7	9.1	0	9.0	0.8	15	1,985	5,722	3,254	2,764	447	3,266	10
60	0.2	17.9	7.5	0	7.9	0.7	16	2,026	5,925	3,232	3,341	542	3,942	10
60	0.5	14.4	5.8	0	6.6	0.6	15	1,759	5,681	1,452	1,458	239	1,724	10
70	0.0	21.3	9.2	0	8.8	1.0	36	2,584	8,104	6,272	3,234	535	3,835	8
70	0.2	16.8	7.1	0	7.4	0.7	18	2,107	6,329	7,696	6,825	1,221	8,177	9
70	0.5	14.8	6.2	0	6.9	0.6	24	2,234	7,001	6,388	6,678	1,180	7,961	9
		17.8	7.5	0	7.9	0.7	17	1,938	5,807	3,528	2,910	498	3,462	86

Table 7. Data Set 4 (Instances with $B = 100$)

		Cplex 12.5			MR									
n	θ	LB0	$\overline{\text{LB0}}$	Opt	LB2	$\overline{\text{LB2}}$	$T_{\overline{LB2}}$	Cut	Col	Nd	T_{MP}	T_{Pr}	T_{Tot}	Opt
40	0.0	25.6	12.7	0	13.2	1.3	21	2,589	5,853	1,544	1,926	382	2,340	10
40	0.2	21.4	10.9	0	10.9	1.0	16	2,345	5,410	616	661	158	830	10
40	0.5	16.3	7.0	0	8.1	0.5	13	1,976	5,115	167	167	43	213	10
50	0.0	25.3	12.3	0	12.3	1.2	46	3,462	8,019	5,311	9,304	1,853	11,290	10
50	0.2	21.1	9.7	0	10.6	0.9	34	2,971	7,255	2,711	5,211	1,237	6,502	10
50	0.5	16.5	7.4	0	8.2	0.8	25	2,478	6,572	1,364	2,259	570	2,860	10
60	0.0	25.2	12.6	0	12.3	1.3	99	4,478	10,694	9,427	15,346	2,940	18,483	6
60	0.2	20.5	9.5	0	10.0	0.9	67	3,651	9,334	6,060	13,896	3,314	17,394	10
60	0.5	16.6	7.8	0	8.5	0.9	47	3,263	8,576	5,586	11,730	3,042	14,909	9
		20.9	10.0	0	10.5	1.0	41	3,024	7,425	3,643	6,722	1,504	8,313	85

The last line of Tables 1–7 indicates the average value of each column (expect for columns labeled Opt where the total number of instances solved to optimality is reported).

Algorithm MR does not require any special parameter tuning. The only parameter involved was the tolerance to indicate when an (AQ) Equation (21) is violated by the MP solution $(\xi, \bar{\xi})$; we set such a tolerance equal to 0.2.

In Table 1, we report a detailed computational comparison of the performance of the four exact algorithms on Data Set 1 instances. All four methods were able to solve all 13 instances to optimality. Algorithms RBM and MR show similar performances and proved to be superior, in terms of computing times, to Cplex 12.5 and AA. By comparing columns LB0, LB1, and LB2, we can see that the linear relaxation of the new formulation F2 leads to much tighter lower bounds than the linear relaxations of formulations F0 and F1. Furthermore, the lower bound $\overline{\text{LB2}}$ is, on average, better than the lower bound $\overline{\text{LB1}}$. Yet we note that, on three instances (i.e., 4×64, 10×10c, and 10×26), $\overline{\text{LB1}}$ is strictly better than $\overline{\text{LB2}}$. This is because of the effectiveness of the set covering inequalities (20) on these few instances; nonetheless, as the additional inequalities (20) worsen the average performance of MR on the other three classes of instances and we wanted to have a single setting on all instances, we did not separate such inequalities on Data Set 1 instances,either.

Table 2 compares the performance of MR with those of Cplex, AA, and RBM on Data Set 2. For each of the three classes of instances, the table reports average values over 10 instances (detailed computational results are reported in Tables EC.8–EC.10 in the online appendix).

Algorithms Cplex, RBM, and MR could solve all 30 instances to optimality, whereas AA could not solve three of them within the imposed time limit of 600 seconds. Algorithm MR is clearly the most performing of the four algorithms. This is mainly because of the lower bounds computed at the root node that are, on average, 0.1% far from optimality.

Tables 3 and 4 compare the performance of Cplex 12.5, RBM, and MR on Data Set 3. Specifically, Table 3 summarizes the results on the nine classes featuring $B = 20$, while Table 4 summarizes the results on the nine classes featuring $B = 50$. On all 180 instances a time limit of 10,800 seconds was imposed on any of the algorithms. Detailed computational results can be found in Tables EC.11–EC.28 in the online appendix.

Table 3 shows that MR closed all instances, RBM closed all but one instance, while Cplex 12.5 could solve only 18 of the 90 instances. In spite of slightly better average lower bounds computed at the root node (see columns $\overline{\text{LB2}}$ and $\overline{\text{LB1}}$), MR is about six times faster than RBM. Note also that the lower bound LB2 given by the linear relaxation of LF2 is almost as good as the lower bounds computed by Cplex 12.5 at the root node.

Table 4 shows that all 90 instances were closed by MR, 16 more than RBM, and 69 more than Cplex 12.5. Even though these instances contain fewer sources and sinks than the instances of Table 3, they are a bit more difficult and the average lower bounds provided by $\overline{\text{LB2}}$ are slightly worse (i.e., on average 0.8% from optimality). Note that around 85% of the computing time spent by MR is spent while solving MP; this same behavior is confirmed on all classes of instances.

In Table 5, we summarize the results achieved by Cplex 12.5 and MR on the 60 instances of Data Set 4 featuring $B = 20$ with a time limit of 12 hours. The detailed results can be found in Tables EC.29–EC.34 in the online appendix. Algorithm Cplex 12.5 could not solve any of the instances, while all but one instance were solved by MR thanks to the small average gap (0.3%) left by $\overline{\text{LB2}}$. Note that $\overline{\text{LB2}}$ leaves a gap similar to the gap left on instances of Table 3 (which are instances with the same features but fewer sources and sinks), showing the effectiveness of the bounding procedure described in Section 5.3; nonetheless, the total number of nodes and the total computing times are much higher.

Table 6 summarizes the results achieved by Cplex 12.5 and MR on the 90 instances of Data Set 4 featuring $B = 50$ with a time limit of 12 hours. The detailed results can be found in Tables EC.35–EC.43 in the online appendix. Of the 90 instances, 86 were close by MR.

Table 7 summarizes the results achieved by Cplex 12.5 and MR on the 90 instances of Data Set 4 featuring $B = 100$ with a time limit of 12 hours. The detailed results can be found in Tables EC.44–EC.52 in the online appendix. Of the 90 instances, 85 were close by MR.

8. Conclusions

In this paper, we proposed a new integer programming formulation of the FCTP having exponentially many variables and a polynomial number of constraints. Variables represent flow patterns from sources to sinks and flow patterns from sinks to sources. The constraints match together the two types of patterns to provide a valid FCTP solution.

A new family of equations that strengthen the linear relaxation of the new formulation are introduced. It is shown that a class of inequalities introduced by Roberti, Bartolini, and Mingozzi (2015) corresponds to different types of surrogate constraints of this new family of equations. The resulting linear relaxation provides stronger lower bounds than those achieved by Roberti, Bartolini, and Mingozzi (2015). The new lower bound is embedded into an exact branch-and-cut-and-price algorithm to achieve an optimal FCTP integer solution.

Computational results on benchmark instances from the literature show that the proposed exact algorithm outperforms the previous exact algorithms from the literature as well as the method of Roberti, Bartolini, and Mingozzi (2015). The new algorithm is several times faster, can solve all instances previously unsolved, and can solve much harder instances with up to 120 sources and 120 sinks in reasonable computing times.

Acknowledgments

The authors acknowledge Enrico Bartolini for his valuable comments and suggestions on the original version of the manuscript. The authors are also grateful to the three anonymous referees for their insightful comments that improved the presentation of the paper.

Appendix

In the following, we briefly describe the inequalities of Class 1 used by Roberti, Bartolini, and Mingozzi (2015) to improve relaxation LF1:

• *Extended Generalized Upper Bound Cover inequalities* (*EGUBC*). These inequalities extend the well-known *generalized upper bound cover* (GUBC) inequalities to the FCTP; see Wolsey (1998); Gu, Nemhauser, and Savelsbergh (1998, 1999).

For a given sink $j \in T$, let $\mathcal{P}_j = \{\xi_l \in \{0,1\}, l \in \mathcal{W}\colon \sum_{l\in\mathcal{W}} w_j^l \xi_l = b_j$ and $\sum_{l\in\mathcal{W}_i} \xi_l \leq 1,\ i \in S\}$ be the set of all integer solutions of the corresponding constraint (7) and all constraints (8) where the "equal to" is replaced with "less than or equal to." A *minimal* GUBC of $\mathcal{P}_j$ is any subset $C \subseteq \mathcal{W}$ such that (i) $\sum_{l\in C} w_j^l > b_j$, (ii) $|C \cap \mathcal{W}_i| \leq 1$, $i \in S$, and (iii) $\sum_{l\in C\setminus\{r\}} w_j^l < b_j$, $\forall\, r \in C$. Let $\mathscr{C}_j$ be the set of all minimal GUBCs of the set $\mathcal{P}_j$. Any F1 solution satisfies the GUBC inequality

$$\sum_{l\in C} \xi_l \leq |C| - 1, \quad C \in \mathscr{C}_j\ j \in T. \tag{A.1}$$

Roberti, Bartolini, and Mingozzi (2015) proposed the following lifting of inequalities (A.1). For each $C \in \mathscr{C}_j$, $j \in T$, let $\rho(C) = \max_{l\in C}\{w_j^l\}$, and let $\gamma_i(C)$, $i \in S$, be a coefficient defined as $\gamma_i(C) = w_j^l$ if $C \cap \mathscr{W}_i = \{l\}$, and $\gamma_i(C) = \rho(C)$ if $C \cap \mathscr{W}_i = \varnothing$. Then, the following EGUBC inequalities are valid for LF1

$$\sum_{i\in S}\sum_{l\in \mathscr{W}_i:\, w_j^l \ge \gamma_i(C)} \xi_l \le |C| - 1, \quad C \in \mathscr{C}_j\ j \in T. \tag{A.2}$$

• *Couple inequalities* (*CPL*). Consider a solution $\boldsymbol{\xi} \in \mathscr{P}_j$, a source $i \in S$, and a sink $j \in T$. If $\boldsymbol{\xi}$ contains a variable $\xi_h = 1$, $h \in \mathscr{W}$, such that $b_j/2 < w_j^h < b_j$ and a variable $\xi_k = 1$, $k \in \mathscr{W}_i$, such that $(b_j - w_j^h)/2 < w_j^k < b_j - w_j^h$, then $\boldsymbol{\xi}$ must contain a variable $\xi_s = 1$, $s \in \mathscr{W}$, such that $0 < w_j^s \le b_j - w_j^h - w_j^k$.

Let $\mathscr{Q} = \{(i, j, q_1, q_2)\colon i \in S,\ j \in T,\ b_j/2 < q_1 < b_j,\ (b_j - q_1)/2 < q_2 < b_j - q_1,\ q_1, q_2 \in \mathbb{Z}_+\}$. The CPL inequalities are defined as

$$\sum_{l\in\mathscr{W}:\, w_j^l \le b_j - q_1 - q_2} \xi_l - \sum_{l\in\mathscr{W}:\, w_j^l = q_1} \xi_l - \sum_{l\in\mathscr{W}_i:\, w_j^l = q_2} \xi_l \ge -1, \quad (i, j, q_1, q_2) \in \mathscr{Q}. \tag{A.3}$$

• *Feasibility inequalities* (*FSB*). Given a sink $j^* \in T$ and an integer $q^* \in \mathbb{Z}$ such that $b_{j^*}/2 < q^* < b_{j^*}$, any F1 solution $\boldsymbol{\xi}$ that contains a variable ξ_k with $w_{j^*}^k \ge q^*$ must contain at least a variable $\xi_l = 1$ with $w_{j^*}^l \le b_{j^*} - w_{j^*}^k$ such that $\sum_{l\in\mathscr{W}:\, w_{j^*}^l \le b_{j^*} - w_{j^*}^k} w_{j^*}^l \xi_l = b_{j^*} - w_{j^*}^k$. Let $\mathscr{F} = \{(j^*, q^*)\colon j^* \in T,\ b_{j^*}/2 < q^* < b_{j^*},\ q^* \in \mathbb{Z}_+\}$. The FSB inequalities are

$$\sum_{l\in\mathscr{W}:\, w_{j^*}^l \le b_{j^*} - q^*} w_{j^*}^l \xi_l + \sum_{l\in\mathscr{W}:\, q^* \le w_{j^*}^l < b_{j^*}} (w_{j^*}^l - b_{j^*})\xi_l \ge 0, \quad (j^*, q^*) \in \mathscr{F}. \tag{A.4}$$

• *Chvátal–Gomory inequalities* (*CGD* and *CGU*):

$$\sum_{l\in\mathscr{W}} \left\lfloor \frac{w_{j^*}^l}{q^*} \right\rfloor \xi_l \le \left\lfloor \frac{b_{j^*}}{q^*} \right\rfloor, \quad 2 \le q^* < b_{j^*}\ j^* \in T, \tag{A.5}$$

$$\sum_{l\in\mathscr{W}} \left\lceil \frac{w_{j^*}^l}{q^*} \right\rceil \xi_l \ge \left\lceil \frac{b_{j^*}}{q^*} \right\rceil, \quad 2 \le q^* < b_{j^*}\ j^* \in T. \tag{A.6}$$

• *Lifted Chvátal–Gomory inequalities* (*LCGD*). For a given sink $j^* \in T$ and a given q^* such that $2 \le q^* < b_{j^*}$, let $\rho_{q^*j^*} = b_{j^*} - \lfloor b_{j^*}/q^* \rfloor q^* + 1$. The following LCGD inequalities hold:

$$\sum_{l\in\mathscr{W}:\, w_{j^*}^l \ge q^*} \left\lfloor \frac{w_{j^*}^l}{q^*} \right\rfloor \xi_l + \sum_{l\in\mathscr{W}_r:\, \rho_{q^*j^*} \le w_{j^*}^l < q^*} \xi_l \le \left\lfloor \frac{b_{j^*}}{q^*} \right\rfloor, \quad 2 \le q^* < b_{j^*}\ r \in S\ j^* \in T. \tag{A.7}$$

Endnote

[1] http://www.spec.org/benchmarks.html (accessed December 15, 2016).

References

Adlakha V, Kowalski K (2003) A simple heuristic for solving small fixed-charge transportation problems. *Omega* 31(3):205–211.

Agarwal Y (2006) *K*-partition-based facets of the network design problem. *Networks* 47(3):123–139.

Agarwal Y, Aneja Y (2012) Fixed-charge transportation problem: Facets of the projection polyhedron. *Oper. Res.* 60(3):638–654.

Aneja Y (1974) On a class of set covering problems. Unpublished doctoral thesis, Johns Hopkins University, Baltimore.

Buson E, Roberti R, Toth P (2014) A reduced-cost iterated local search heuristic for the fixed charge transportation problem. *Oper. Res.* 62(5):1095–1106.

Fisk J, McKeown PG (1979) The pure fixed charge transportation problem. *Naval Res. Logist. Quart.* 26(4):631–641.

Göthe-Lundgren M, Larsson T (1994) A set covering reformulation of the pure fixed charge transportation problem. *Discrete Appl. Math.* 48(3):245–259.

Gu Z, Nemhauser GL, Savelsbergh MWP (1998) Lifted cover inequalities for 0-1 integer programs: Computation. *INFORMS J. Comput.* 10(4):427–437.

Gu Z, Nemhauser GL, Savelsbergh MWP (1999) Lifted cover inequalities for 0-1 integer programs: Complexity. *INFORMS J. Comput.* 11(1):117–123.

Hirsch WM, Dantzig GB (1968) The fixed charge problem. *Naval Res. Logist. Quart.* 15:413–424.

Hultberg TH, Cardoso DM (1997) The teacher assignment problem: A special case of the fixed charge transportation problem. *Eur. J. Oper. Res.* 101(3):463–473.

Ortega F, Wolsey LA (2003) A branch-and-cut algorithm for the single commodity, uncapacitated, fixed-charge network flow problem. *Networks* 41(3):143–158.

Rardin RL, Wolsey LA (1993) Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. *Eur. J. Oper. Res.* 71(1):95–109.

Roberti RE, Bartolini A, Mingozzi A (2015) The fixed charge transportation problem: An exact algorithm based on a new integer programming formulation. *Management Sci.* 61(6):1275–1291.

Stroup JW (1967) Allocation of launch vehicles to space missions: A fixed-cost transportation problem. *Oper. Res.* 15(6):1157–1163.

Walker WE (1976) A heuristic adjacent extreme point algorithm for the fixed charge problem. *Management Sci.* 22(5):587–596.

Wolsey LA (1998) *Integer Programming* (Wiley-Interscience, New York).